## Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

## Listing of Claims:

1-3 (Canceled)

4. (Previously Presented)   A JAVA™ virtual machine residing on a computing apparatus and operating in a JAVA™ computing environment, said JAVA™ virtual machine capable of executing a Bytecode instruction to determine a string representation associated with a JAVA™ object, thereby determining said string representation of said JAVA™ object without invoking a JAVA™ "to_string" method, wherein said virtual machine is capable of performing the following operations when said Bytecode instruction is executed in order to determine said string representation of a said JAVA™ object:

   popping a reference to said JAVA™ object from an execution stack;

   accessing a field of said JAVA™ object by using said reference to said JAVA™ object in order to obtain data representing said field;

   determining a string representation for said field of said JAVA™ object after said accessing of said field of said JAVA™ object by using said reference to said JAVA™ object stored on said execution stack; and

   pushing on said execution stack a reference to said string representation after said determining of said string representation.

5-6. (Canceled)

7. (Previously Presented)   A JAVA™ virtual machine as recited in claim 4, wherein said JAVA™ virtual machine operates in an embedded system.

8. (Currently Amended)  In a JAVA<sup>TM</sup> computing environment, a method of retrieving by a virtual machine a string representation for a JAVA<sup>TM</sup> object, said virtual machine residing on a computing apparatus, said method comprising:

receiving a JAVA<sup>TM</sup> Bytecode instruction in a stream of JAVA<sup>TM</sup> Bytecodes suitable for execution by a virtual machine operating in said JAVA<sup>TM</sup> computing environment, wherein said JAVA<sup>TM</sup> Bytecode instruction is designated to determine said string representation for said JAVA<sup>TM</sup> object;

pushing a reference to said JAVA<sup>TM</sup> object on an execution stack;

executing said JAVA<sup>TM</sup> Bytecode instruction;

popping said reference to said JAVA<sup>TM</sup> object from said execution stack;

accessing a field of said JAVA<sup>TM</sup> object by using said reference to said JAVA<sup>TM</sup> object in order to obtain data representing said field;

determining a string representation for said field of said JAVA<sup>TM</sup> object by after said accessing of said field of said JAVA<sup>TM</sup> object by using said reference to said JAVA<sup>TM</sup> object stored on said execution stack; and

pushing on said execution stack a reference to said string representation after said determining of said string representation, thereby allowing said string representation to be determined without invoking a JAVA<sup>TM</sup> method.

9-10. (Cancelled)

11. (Previously Presented)  A method as recited in claim 8, wherein said pushing of a reference to said JAVA<sup>TM</sup> object is performed by execution of a JAVA<sup>TM</sup> Aload execution.

12. (Previously Presented)   A method as recited in claim 11, wherein said method is performed by a virtual machine.

13. (Previously Presented)   A method as recited in claim 12, wherein said virtual machine is operating in an embedded system.

14. (Previously Presented)   A computer readable medium including computer program code for retrieving a string representation for a JAVA$^{TM}$ object, said computer readable medium comprising:

computer program code for receiving a JAVA$^{TM}$ Bytecode instruction in a stream of JAVA$^{TM}$ Bytecodes suitable for execution by a virtual machine operating in a JAVA$^{TM}$ computing environment, and

wherein said JAVA$^{TM}$ Bytecode instruction operates to determine said string representation associated with said JAVA$^{TM}$ object by using a reference to said JAVA$^{TM}$ object stored on an execution stack, thereby allowing said string representation to be determined without invoking a JAVA$^{TM}$ method.

15. (Previously Presented)   A computer readable medium as recited in claim 14, wherein said computer readable medium further comprises:

computer program code for popping a reference to a JAVA$^{TM}$ object from an execution stack;

computer program code for determining a string representation of a field associated with said JAVA$^{TM}$ object; and

computer program code for pushing a reference to said string representation of said field on top of said execution stack.

16. (Canceled)

17. (Previously Presented)   A computer readable medium as recited in claim 15, wherein said computer program code comprises a JAVA™ Aload instruction that when executed performs the pushing of said reference.

18. (Previously Presented)   A computer readable medium as recited in claim 17, wherein said computer readable medium is read by a JAVA™ virtual machine.

19. (Previously Presented)   A computer readable medium as recited in claim 18, wherein said virtual machine is operating in an embedded system.

20. (Previously Presented)   A computer system for retrieving a string representation for a JAVA™ object in a JAVA™ computing environment, said computer system capable of operating to:

receive a JAVA™ Bytecode instruction in a stream of JAVA™ Bytecodes suitable for execution by a virtual machine operating in said JAVA™ computing environment, wherein said JAVA™ Bytecode instruction operates to determine said string representation associated with said JAVA™ object, thereby allowing said string representation to be determined without invoking a JAVA™ method;

push a reference to said JAVA™ object on an execution stack;

execute said JAVA™ Bytecode instruction;

pop said reference to said JAVA™ object from said execution stack;

access a field associated with said JAVA™ object by using said reference;

determine a string representation of said field by accessing said field using said reference; and

push a reference to said string representation of said field on top of said execution stack.

21. (Previously Presented)   A computer system as recited in claim 20, wherein said pushing of a reference to said JAVA<sup>TM</sup> object is performed by execution of a JAVA<sup>TM</sup> Aload bytecode.

22. (Previously Presented)   A computer system as recited in claim 21, wherein said virtual machine operates in an embedded system.

23. (Previously Presented)   A virtual machine as recited in claim 4, wherein said reference to said JAVA<sup>TM</sup> object is stored on said execution stack by executing another Bytecode instruction.

24. (Previously Presented)   A virtual machine as recited in claim 23, wherein said other Bytecode instruction is a JAVA<sup>TM</sup> Aload bytecode instruction.